Overview

To get started with RunPod:

- Create a RunPod account
- Add funds
- Use the RunPod SDK to build and connect with your Serverless Endpoints

Quick migration with Docker

The rest of this guide will help you set up a RunPod project.

Setting up your project

Just like with Banana, RunPod provides a Python SDK to run your projects.

To get started, install setup a virtual environment then install the SDK library.

macOS Windows

Create a Python virtual environment with venv:

```
python3 -m venv env
source env/bin/activate
```

To install the SDK, run the following command from the terminal.

```
python -m pip install runpod
```

Project examples

RunPod provides a repository of templates for your project.

You can use the template to get started with your project.

```
gh repo clone runpod-workers/worker-template
```

Now that you've got a basic RunPod Worker template created:

- Continue reading to see how you'd migrate from Banana to RunPod
- See Generate SDXL Turbo for a general approach on deploying your first Serverless Endpoint with RunPod.

Project structure

When beginning to migrate your Banana monorepo to RunPod, you will need to understand the structure of your project.

Banana **RunPod**

Banana is a monorepo that contains multiple services. The basic structure for Banana projects is aligned with the RunPod Serverless projects for consistency:

```
.
├── Dockerfile              # Docker configuration
├── README.md               # Project documentation
├── banana_config.json      # Configuration settings
├── requirements.txt        # Dependencies
└── src
    ├── app.py              # Main application code
    └── download.py         # Download script
```

Both project setups at a minimum contain:

- `Dockerfile`: Defines the container for running the application.
- Application code: The executable code within the container.

Optional files included in both setups:

- `requirements.txt`: Lists dependencies needed for the application.

Banana Configuration settings

Banana configuration settings are stored in a `banana_config.json` file.

Banana uses a `banana_config.json` file which contains things like Idle Timeout, Inference Timeout, and Max Replicas.

Idle Timeout

RunPod allows you to set an Idle Timeout when creating the Endpoint. The default value is 5 seconds.

Inference Timeout

RunPod has a similar concept to Inference Timeout. For runs that are take less than 30 seconds to execute, you should use the `run_sync` handler. For runs that take longer than 30 seconds to execute, you should use the `sync` handler.

Max Replicas

When creating a Worker in RunPod, you can set the max Workers that will scale up depending on the amount of Worker sent to your Endpoint. For more informaiton, see Scale Type

> **NOTE**
>
> When creating a Worker, select the **Flashboot** option to optimize your startup time.

Edit this page

Previous
« Banana

Tutorials

AI APIs

Community

Discord

Contact us

More

Blog

GitHub